

Rec'd PCT/EP 539 813 JUN 2004

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
8 July 2004 (08.07.2004)

PCT

(10) International Publication Number
WO 2004/057538 A2

(51) International Patent Classification⁷: **G06T 11/00**

(21) International Application Number:
PCT/EP2003/013227

(22) International Filing Date:
25 November 2003 (25.11.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
02028537.5 20 December 2002 (20.12.2002) EP
60/436,162 23 December 2002 (23.12.2002) US

(71) Applicant (for all designated States except US): TELEFONAKTIEBOLAGET LM ERICSSON (publ)
[SE/SE]; S-164 83 Stockholm (SE).

(72) Inventors; and

(75) Inventors/Applicants (for US only): LEDFELT, Erik
[SE/SE]; Sidensvansgatan 38, S-235 38 Vellinge (SE).

TOLF, Fredrik [SE/SE]; Stora Nygatan 53, SE-211 37 Malmö (SE). LEVIN, Martin [SE/SE]; Östra Promenaden 23, SE-211 28 Malmö (SE). AKENINE-MÖLLER, Tomas [SE/SE]; Träkilsgratan 137, SE-416 78 GÖTEBORG (SE).

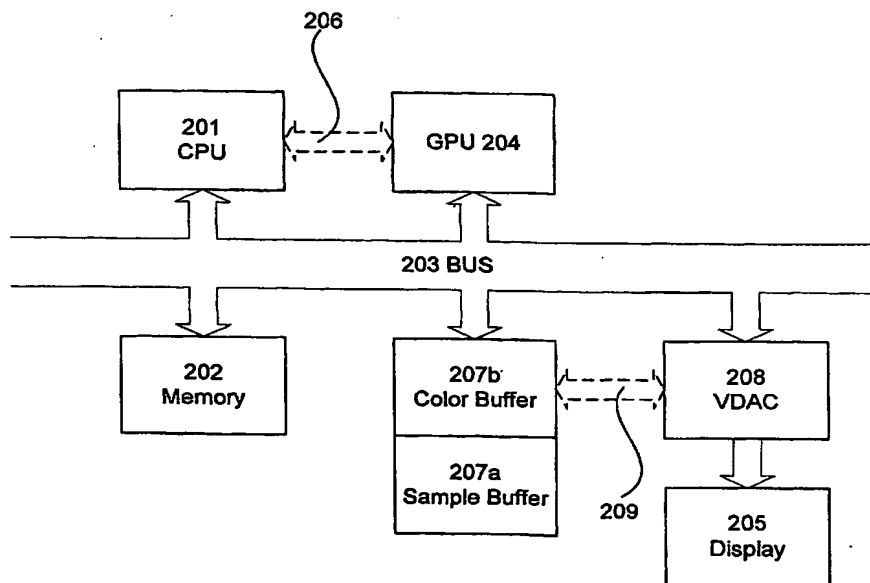
(74) Agent: BYSTRÖM, Linus; Ström & Gulliksson IPC AB, P.O. Box 793, SE-220 07 Lund (SE).

(81) Designated States (national): AE, AG, AL, AM, AT (utility model), AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, EG, ES, FI (utility model), FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK (utility model), SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (regional): ARIPO patent (BW, GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),

[Continued on next page]

(54) Title: LOW-COST SUPERSAMPLING RASTERIZATION



(57) Abstract: A sampling pattern covering an array of pixels for use in an anti-aliasing system is disclosed where each, pixel has a pattern of sample points at the edges of the pixel. Moreover is the sample point pattern of each pixel a mirror image and different from the pattern of a directly neighboring pixel

WO 2004/057538 A2



European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

- *without international search report and to be republished upon receipt of that report*

Low-cost supersampling rasterization

Technical Field

Generally, the present invention relates to graphic
5 processing and more specifically to a method and apparatus
for producing high-quality anti-aliased graphic pictures at
high frame rates with low computational cost.

Description of the Prior Art

10 Since the early days of computer graphics, aliasing
has been a problem when presenting still or moving pictures
on a display.

One approach to combat the low visual quality of
aliased pictures is to use what is known as supersampling.
15 Supersampling will provide a good picture quality but has
the drawback of a low frame rate due to a heavy
computational burden. More specifically, supersampling
renders a picture at a higher resolution than the final
resolution that is displayed on the screen. This is done by
20 rendering multiple sub-pixel samples for each pixel to be
displayed, i.e. the value of each pixel will be a weighted
sum of the sub-pixel sample values. For example may each
displayed pixel comprise the filtered, weighted sum of a
group of four sub-pixel samples inside a pixel. As can be
25 readily understood, this implies that the graphics hardware
has to process four times as many samples for each
displayed pixel.

The patent document WO-00/33256 discloses a system
that utilizes a supersampling scheme. Each pixel is divided
30 into a more or less fine-meshed grid which defines a sub-
pixel grid, where sample points may be located. The sub-
pixel sample points may be arranged in many different
configurations inside the pixel boundaries. The sample
point configuration pattern is then repeated for every
35 pixel to be rendered. The final value for each pixel

comprises the weighted sum of three or more samples located in sub-pixels according to the discussion above.

The patent document US-6,057,855 discloses a system for determining a value for an anti-aliased pixel. In ,
5 similarity to the above document, the system uses a supersampling scheme for determining the sample values at multiple sub-pixel sample locations.

In order to lower the computational burden for producing anti-aliased pixels, a modified supersampling
10 scheme may be used. The key idea of this supersampling scheme is to place the sub-pixel sample locations in such positions so that the value of one or more of the sample locations may be used for calculating the final value for more than one pixel. A supersampling scheme of this kind is
15 also referred to as a sample-sharing scheme.

The GeForce3 graphics processing unit from NVIDIA Corporation, Santa Clara, USA provides hardware that supports supersampling and sharing of sub samples between pixels. The supersampling scheme is referred to as
20 "Quincunx" and presents a sub-pixel sample pattern in form of a "5" on a die, i.e. five sub-pixel samples are used for calculating the value of the final pixel. However, due to the placing of the sample locations, only two samples per pixel need to be calculated; the rest of the sample values
25 are obtained from the neighboring pixels. The center sub-pixel sample is given the weight 0.5 while the peripheral sub-pixel samples are given the weight 0.125 each. In a subsequent step, the sub-pixel samples are filtered in the same way as with an ordinary supersampling scheme.

30 Detailed information regarding the Quincunx scheme is found in "Technical Brief, HRAA: High-Resolution Antialiasing through Multisampling" from NVIDIA Corporation. This document is e.g. retrievable from the NVIDIA Corporation web site "www.nvidia.com".

The number of gray levels between black and white (in a monochrome scheme) depends on how many sub-pixel sample points that are used. In case four sub-pixel sample locations are used, there will at best be three gray shades
5 between black and white. Consequently, the Quincunx scheme above will at best provide four shades of gray. However, as will be discussed below, the effective number of gray shades for the Quincunx scheme may be as low as two.

Accordingly, the computational burden for producing
10 anti-aliased pixels is a problem in modern computer graphics systems. As is easily understood, the problem becomes even bigger if an anti-aliasing scheme is to be used for producing moving pictures on a device with reduced computational capability, such as a mobile telephone or a
15 PDA (personal digital assistant).

Summary of the Invention

The present invention seeks to provide a method and apparatus for producing high-quality anti-aliased pictures
20 at a low computational cost.

This object has been achieved by a sampling pattern covering an array of pixels, where each pixel has a pattern of sample points at the edges of the pixel, and where the sample point pattern of each pixel is a mirror image and
25 different from the pattern of a directly neighboring pixel.

Brief Description of the Drawings

A preferred embodiment of the present invention will now be described with reference to the accompanying
30 drawings, in which:

FIG. 1 is a schematic block diagram illustrating a graphics system for creating anti-aliased pictures,

FIG. 2 is a schematic drawing illustrating the calculation of the sub-pixel sample locations according to
35 the present invention,

FIG. 3 is a schematic illustration of a mirroring step according to a preferred embodiment of the present invention,

FIG. 4 is another schematic illustration of a
5 mirroring step according to a preferred embodiment of the present invention

FIG. 5 is a schematic flow chart illustrating the method for producing anti-aliased pictures according to the present invention, and

10 FIG. 6 is a schematic drawing illustrating the calculation of pixel values according to the present invention compared to a prior art scheme.

FIG 7 is a graphic comparison between no anti-aliasing, a prior art scheme and the anti-aliasing scheme
15 according to the present invention.

Detailed Disclosure of a Preferred Embodiment

Figure 1 is a block diagram of an example of a system for drawing lines or polygons. A CPU (Central Processing
20 Unit) 201 is connected to a memory 202 by means of a data bus 203. The memory 202 comprises the application program that is run on the system, e.g. a computer game or a CAD (Computer Aided Design) program. As with most computer systems, the CPU 201 fetches instructions in the memory 202
25 and executes them in order to perform specific tasks. In this context, a task for the CPU 201 is to provide a GPU 204 (Graphics Processing Unit) with information regarding the objects that shall be drawn on a display 205. It is emphasized that the GPU 204 may be in form of a processor, such as a DSP (Digital Signal Processor), or in form of an
30 ASIC (Application Specific Integrated Circuit), FGPA (Field-Programmable Gate Array), hard-wired logic etc, or it may be executed on the CPU 201. The GPU 204 is also connected to the bus 203 but may as well be connected to
35 the processor by means of a separate high-speed bus 206 in

case a lot of information is to be transferred between the CPU 201 and the GPU 204. The data transfers on the separate high-speed bus 206 will then not interfere with the data traffic on the ordinary bus 203.

5 Moreover, a display memory 207 is also connected to the bus 203 and stores information sent from the GPU 204 regarding the pictures (frames) that shall be drawn on the display 205. More specifically, the display memory contains a sample buffer 207a and a color buffer 207b. As will be
10 discussed below, according to the present invention, the sample buffer 207a contains approximately twice as many samples as there are pixels in the final color buffer 207b. The color buffer 207b holds the colors of the pixels to be displayed on screen after the rendering of an image is
15 complete. As with the interconnection between the CPU 201 and the GPU 204, the display memory 207 may be connected directly to the GPU 204 by means of a separate, high-speed bus. Since the GPU 204 and the display memory 207 normally are used for producing moving images, it is preferred that
20 the link between these two units is as fast as possible and does not block the normal traffic on the bus 203.

 The display memory 207 is connected to a VDAC 208 (Video Digital to Analog Converter), either by means of the shared bus 203 or by a separate high-speed bus 209, which
25 reads the information from the color buffer 207b and converts it to an analog signal, e.g. a RGB (Red, Green, Blue) composite signal, that is provided to the display 205 in order to draw the individual pixels on the screen.

 As discussed above, many different techniques have
30 been used in order to produce anti-aliased representations of lines and polygons. As can be seen in figure 2a and 2b, the present invention uses a variant of a super-sampling scheme. As with the Quincunx scheme, the sub-pixel sample locations 303-306 are placed at the edges of the pixel 301,
35 302. As discussed above, this allows for sample sharing

between different pixels 301, 302 in the display memory
207.

However, the samples are not placed in the corners of
the pixel 301, 302 as with the Quincunx scheme. Instead, in
5 a preferred embodiment, one sub-pixel sample location is
defined for each edge of the pixel 301, 302 in a rotated
square-shaped configuration and is given a weight of 0.25
each. This is explained in figure 2a and 2b by
superimposing a grid over the pixel 301, 302 and defining a
10 possible sample point wherever the grid intersects an edge
of a pixel 301, 302. The equations for determining the
precise sub-pixel sample locations are shown under figures
2a and 2b respectively. Alternatively, the edges of the
pixels in the discussion above may be substituted by one or
15 more mirroring planes in case the sampling pattern is
translated in any direction. The mirror planes will then
normally be parallel with the edges of the pixels and with
spacing equal to the distance between the edges of the
pixels. For example, the sampling pattern may be translated
20 a small amount to the left, wherein the sub-pixel sample
locations no longer resides on the edges of the pixels. In
this case it is still possible to define one or more
mirroring planes for creating a sample pattern according to
the present invention. This will become apparent by the
25 discussion below in relation to figure 3.

The placing of the sample locations 303-310 will
break the symmetry of the configuration which will increase
the anti-aliasing effect of near to vertical lines and near
to horizontal lines. To illustrate this, lets assume a near
30 to horizontal edge of a polygon that is drawn on a display
across one or more pixels 301, 302. If e.g. the Quincunx
scheme is used for producing an anti-aliased representation
of the line, four sample points, one in each corner of the
pixel 301, 302, will be used. For some pixels 301, 302, the
35 edge will cover only the top part but will still cover the

two uppermost sub-pixel sample locations. Consequently, by examining the values from the sub-pixel sample locations in the pixel 301, 302, the anti-aliased value of the pixel will be 0.25 even if half the pixel is covered by the edge (i.e. until the line covers the sub-pixel sample in the center). The pixel will hence be incorrectly presented on the screen.

If the same situation applies with the use of the present invention a near to horizontal line, covering a small part of the top of the pixel 301, 302, will not cover both sub-pixel sample locations but only the sample location on the uppermost horizontal edge of the pixel 301, 302 due to the placing of the sample locations. The reason for mirroring the sample locations for every other pixel will be discussed below.

Figure 3 illustrates an important feature of the present invention. In accordance with the above, the sub-pixel sample locations 403-406 of the leftmost pixel 401 are not placed in the corners of the pixel as is the case with the Quincunx scheme. In the following text, this sub-pixel sample configuration will be referred to as "quad A". Correspondingly, a pixel 402 presenting a sub-pixel sample configuration that is a mirror image of "quad A" will be referred to as "quad B". As can be seen in figure 3, the sub-pixel sample locations 406-409 in the rightmost pixel 402 corresponds to the quad B locations according to the above. As mentioned above, by examining the configurations of quad A and quad B side-by-side it is evident that the sub-pixel sample locations 406-409 of quad B is a mirror image of the corresponding locations 403-406 in quad A reflected at the right vertical edge 410 of quad A (and consequently the left vertical edge of quad B).

By mirroring the locations of the sub-pixel sample locations 403-409 it is possible to share the sample 406 between the two pixels and still break up the symmetry of

the configuration and achieve a better anti-aliasing result according to the above. Another important feature is that there is only one sample per row and column. In e.g. Quincunx, there are two samples for the top row.

5 Figure 4 further illustrates the anti-aliasing scheme according to the present invention. The upper left pixel 501 contains four sub-pixel sample points 510-513 in a quad A configuration. The pixel 502 to the right of this pixel 501 also contains four sub-pixel sample locations 513-516
10 in a quad B configuration, which are reflected at the right edge of the leftmost pixel 501. Moreover, a third pixel 503 also contains four sub-pixel sample points 516-519 in a quad A configuration. As can be seen from figure 4, the upper row of pixels 501-503 share one sub-pixel sample
15 location 513, 516 between each pair of pixels 501-502, 502-503.

Next row starts with a pixel 504 presenting a quad B configuration of sub-pixel sample points 511, 520-522. The sample location 511 is shared between this pixel 504 and
20 the pixel 501 on the row above. By examining the configurations of quad A (the topmost pixel 501) and quad B (the lower pixel 504) side-by-side it is evident that the sub-pixel sample locations 511, 520-522 of quad B is a mirror image of the corresponding locations 510-513 in quad
25 A reflected at the bottom horizontal edge 530 of pixel 501 (and consequently the top horizontal edge of pixel 504).

The next pixel 505 on the second row contains four sub-pixel sample points 515, 522-524 in a quad A configuration. What is important to notice is that this
30 pixel 505 share one sample point 515 with the pixel 502 on the row above and one sample point 522 with the pixel 504 to the left. The same applies to the rightmost pixel 506 on the second row, which also shares two sample points 517, 524 with the neighboring pixels 503, 505.

Consequently, by using the mirroring scheme of the present invention, all pixels, except for the uppermost and leftmost pixels 501-504 on a display 205, require a calculation of only two new sub-pixel sample location

5 values when determining the final value of the pixels 501-506. Alternatively, all pixels except the rightmost column and the bottommost row require only two samples.

The sample locations in the pixels may be traversed by scanning the lines from left to right. Alternatively,
10 the scanning direction may be altered every other line in order to render the memory usage more effective. It is understood that any traversal scheme can be implemented in conjunction with the supersampling scheme according to the present invention.

15 Figure 5a is a flow chart illustrating a method for producing high-quality anti-aliased pictures according to a preferred embodiment of the present invention. In step 610 the CPU runs the application program (e.g. a computer game) and generates the 3D objects (normally polygons in form of
20 triangles) that shall be converted into a 2D-presentation on the display.

Next, in step 620, the CPU or the GPU/hardware calculates the different visual effects that affect the appearance of the object on the display, such as lighting,
25 clipping, transformations, projections, etc. As triangles are normally used when creating 3-D objects in computer graphics, the pixel coordinates of the vertices of the triangles are finally calculated.

In step 630 the CPU or the GPU/hardware interpolates
30 texture coordinates over the polygon in order to ensure that a correct projection is obtained. In addition to this the CPU or GPU/hardware may also interpolate one or more colors, another set of texture coordinates, fog, and more. It also performs Z-buffer tests, and ensures that the final
35 pixel obtains the correct color.

Figure 5b is a more detailed flow chart illustrating step 630 in figure 5a. To increase the intelligibility of the flow chart in figure 5b, references are also made to figure 6a. Step 631 is a polygon (triangle) setup stage
5 where the CPU or the GPU/hardware calculates interpolation data that is used over the entire polygon 801.

A scan conversion is performed in step 632, wherein the CPU or the GPU/hardware identifies pixels 703 or sample points 704 that lie inside the boundaries 705 of the
10 polygon 701. There are many different ways to perform this identification. A simple approach is to scan the horizontal rows one by one.

All visible sample points 704 are transferred to step 633 which calculates the color of each visible pixel 701 by
15 means of the textures and the interpolated color(s). The color of each sample is written to the sample buffer 207a. After all polygons have been processed, the sample buffer 207a will contain the picture in a high-resolution format (2 samples pixel of the final image). Only visible samples
20 are processed in this stage. Samples that are not visible, i.e. samples that are behind a previously drawn polygon, will not contribute to the final picture. In a final stage, the samples are filtered to produce a picture of correct size. More specifically, four samples per pixel will be
25 averaged to form the final pixel color stored in the color buffer 207b.

With reference to figures 6a and 6b, a comparison will now be made between the Quincunx scheme and the scheme according to the present invention. The sub-pixel sampling
30 pattern according to the present invention is illustrated in figure 6a, and the sub-pixel sampling pattern according to the Quincunx scheme is illustrated in figure 6b.

Assume that the inside of a triangle is colored white (encoded as 1.0), and the outside colored black (encoded as
35 0.0). Anything in between 0.0 and 1.0 represents a gray

scale. Also, it should be noted that the same applies as well to colors or any other representation. As can be seen from the figures of this example, a polygon, in this case a triangle, is covering a 6 x 6 pixel matrix. However, the number of pixels are not restricted to this number and depends on the specific application, i.e. a desktop computer system will use a higher resolution (more pixels) than e.g. a mobile telephone. The same working principle applies to any system irrespective of the resolution of the system. In both figure 6a and 6b, pixels that are completely inside the triangle will obtain the value 1 (completely white). In the Quincunx scheme, this arises from the summing-up of the corner samples (each with the weight 0.125) and the center sample (with weight 0.5). Correspondingly, in the scheme according to the present invention, the same value arises from the summing-up of the four edge sampling positions (each with the weight 0.25).

In figure 6a, the leftmost column will obtain the values (from top to bottom): 0.25, 0.5, 0.5, 0.5, 0.5, and 0.25, where each number represents a gray scale color. That is, the vertices of the triangle will have a slightly darker shade of gray than the central part of the left edge of the triangle.

Correspondingly, in figure 6b, the leftmost column will obtain the values: 0.125, 0.75, 0.75, 0.25, 0.25, and 0.125. What is important is the abrupt jump between the third and fourth pixel in the column. As mentioned above, the calculated pixel values for a near to vertical line will always make an abrupt jump from 0.25 to 0.75 when the Quincunx scheme is used, even though it is theoretically possible to obtain a value of 0.375, 0.5, and 0.625. On the other hand, the mirroring scheme according to the present invention will give a smoother transition between the different possible pixel values.

Aliasing is very noticeable when drawing almost vertical lines and almost horizontal lines, and thus it is important that the anti-aliasing scheme produces good result when edges are near to vertical or near to
5 horizontal.

The above reasoning is further illustrated in figures 7a-c, where a comparison between no anti-aliasing 7a, the Quincunx scheme 7b, and the scheme according to the present invention 7c is shown. The figures clearly illustrates that
10 the anti-aliasing effect for both for a near to vertical as well as for a diagonal line is enhanced by the scheme according to the present invention. More specifically, the effective number of gray levels presented by the Quincunx scheme is reduced to two as described above while the
15 scheme according to the present invention presents three levels of gray between black and white.

The present invention has been described above with reference to a preferred embodiment. However, other embodiments than the one disclosed herein are possible
20 within the scope of the invention, as defined by the appended independent claims.

CLAIMS

1. A sampling pattern covering an array of pixels for use in an anti-aliasing system, where each pixel has a pattern of sample points at one or more than one mirror
5 plane within the array of pixels, characterized in that the sample point pattern of each pixel is a mirror image and different from the pattern of a directly neighboring pixel.
- 10 2. The sampling pattern according to claim 1, wherein the mirror planes are located on the edges of the pixel.
3. The sampling pattern according to claim 1 or 2, wherein the pattern has one sample point per pixel mirror
15 plane.
4. The sampling pattern according to claim 1 to 3, wherein the (x, y) coordinates of the sample points for a pixel are related according to (0, a), (a, 1), (b, 0), and
20 (1, b).
5. The sampling pattern according to claim 1 to 3, wherein the (x, y) coordinates of the sample points for a pixel are related according to (0, b), (a, 0), (b, 1), and
25 (1, a).
6. The sampling pattern according to claims 4 or 5, wherein the sum "a+b" is in the range 0,5 - 1,5.
- 30 7. The sampling pattern according to claims 4-6, wherein $a = 1/3$ and $b = 2/3$.
8. The use of a sampling pattern according to any of claims 1-7 in a pixel anti-aliasing system.

9. The use of a sampling pattern according to claim 8 for processing a still image.

10 8 for processing a video sequence.

11. A method for creating a sampling pattern covering an array of pixels for use in an anti-aliasing system, where each pixel has a pattern of sample points at the edges of the pixel, characterized by

defining the sample point pattern of each pixel so that it is a mirror image and different from the pattern of a directly neighboring pixel

15 12. The method according to claim 11, wherein the pattern has one sample point per pixel edge

13. The method according to claim 11 or 12, wherein the (x, y) coordinates of the sample points for a pixel are related according to (0, a), (a, 1), (b, 0), and (1, b).

14. The method according to claim 11 or 12, wherein the (x, y) coordinates of the sample points for a pixel are related according to (0, b), (a, 0), (b, 1), and (1, a).

25 15. The method according to claims 13 or 14, wherein the sum "a+b" is in the range 0,5 - 1,5.

30 16. The method according to claims 13 to 15, wherein a = 1/3 and b = 2/3.

17. An anti aliased image created by processing an image according to any of the steps 11-16

18. An anti-aliasing system comprising a GPU, wherein the GPU is adapted to define a pattern of sample points at the edges of a pixel, **characterized in that**

5 the GPU is adapted to define the sample point pattern of each pixel so that it is a mirror image and different from the pattern of a directly neighboring pixel

19. The system according to claim 18, wherein the GPU is implemented in hardware.

10

20. The system according to claim 18, wherein the GPU is implemented in software.

21. The system according to claims 18 to 20, wherein
15 the (x, y) coordinates of the sample points for a pixel are related according to (0, a), (a, 1); (b, 0), and (1, b).

22. The system according to claims 18 to 20, wherein the (x, y) coordinates of the sample points for a pixel are
20 related according to (0, b), (a, 0), (b, 1), and (1, a).

23. The system according to claims 21 or 22, wherein the sum "a+b" is in the range 0,5 - 1,5.

24. The system according to claims 21 to 23, wherein
25 a = 1/3 and b = 2/3.

25. A computer program product directly loadable into an internal memory associated with a CPU, said CPU being
30 operatively coupled to a GPU for defining a pattern of sample points at the edges of a pixel, comprising program code for

defining the sample point pattern of each pixel so that it is a mirror image and different from the pattern of
35 a directly neighboring pixel

26. A computer program product as defined in claim 22, embodied on a computer-readable medium.

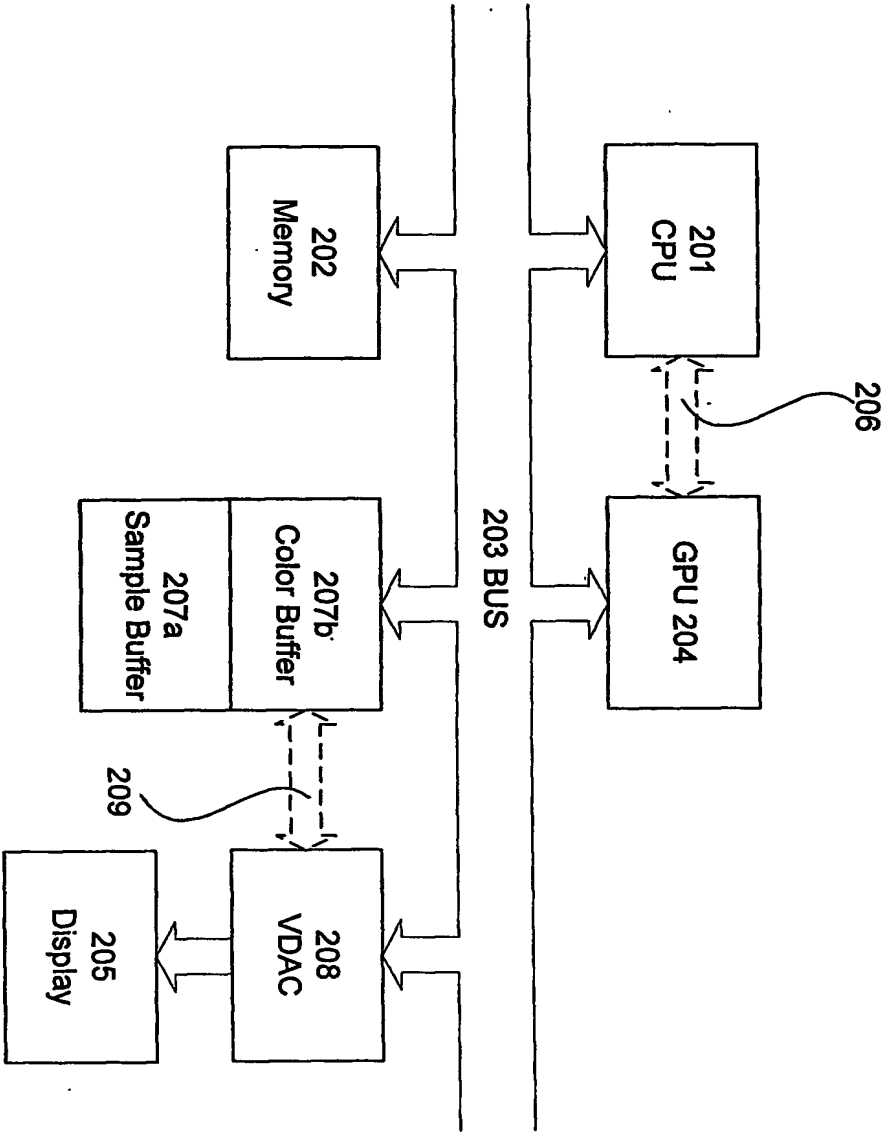
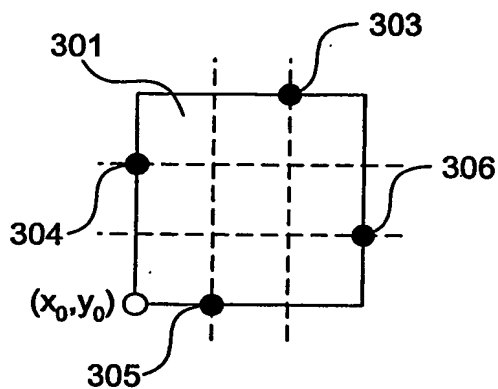


Fig 1



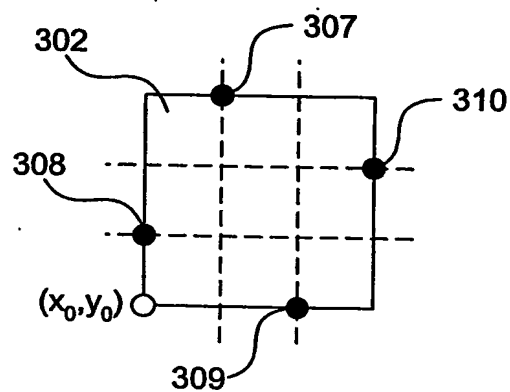
Quad A

$$303 = (x_0 + 2/3, y_0 + 1)$$

$$304 = (x_0, y_0 + 2/3)$$

$$305 = (x_0 + 1/3, y_0)$$

$$306 = (x_0 + 1, y_0 + 1/3)$$

Fig 2a

Quad B

$$307 = (x_0 + 1/3, y_0 + 1)$$

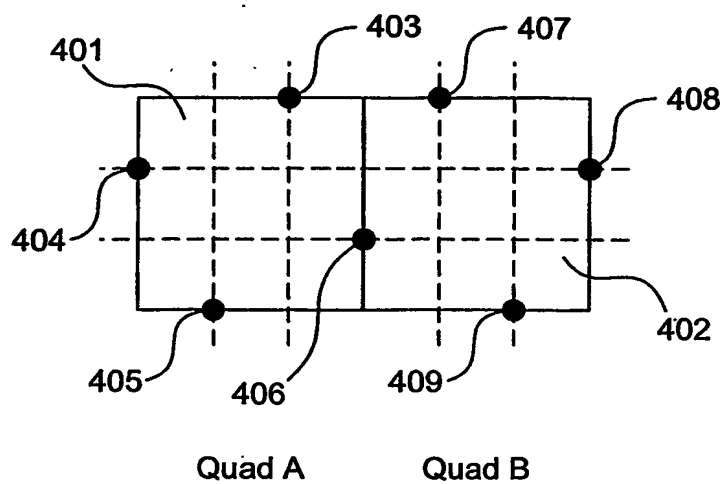
$$308 = (x_0, y_0 + 1/3)$$

$$309 = (x_0 + 2/3, y_0)$$

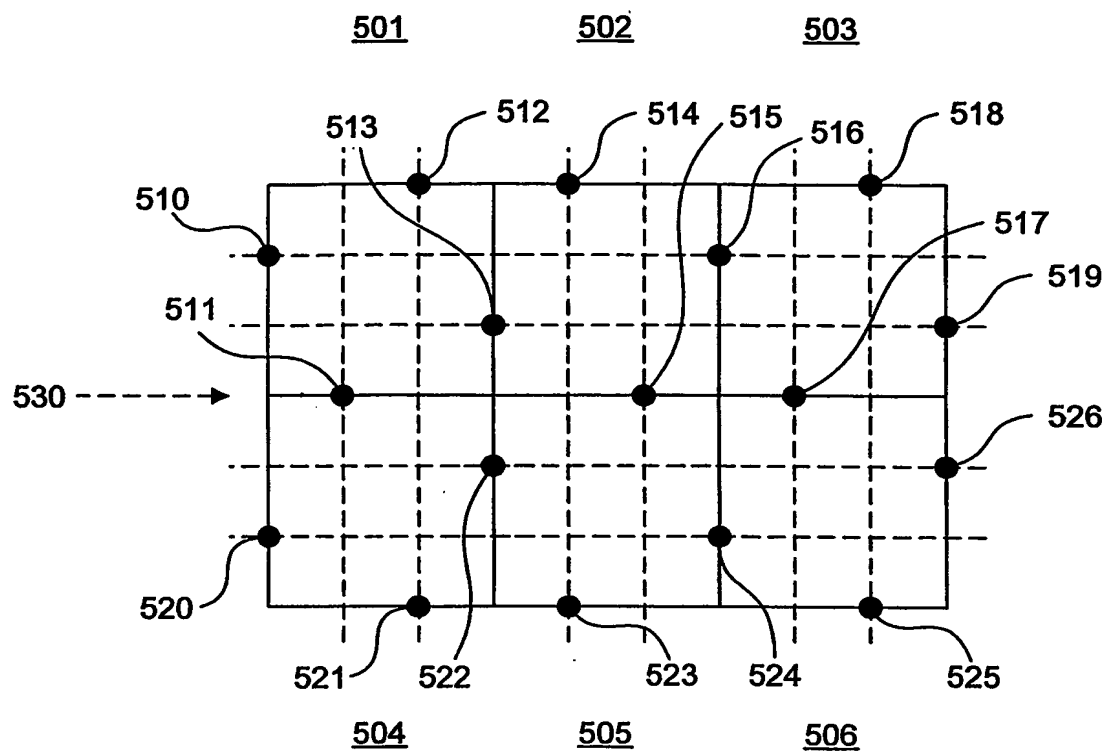
$$310 = (x_0 + 1, y_0 + 2/3)$$

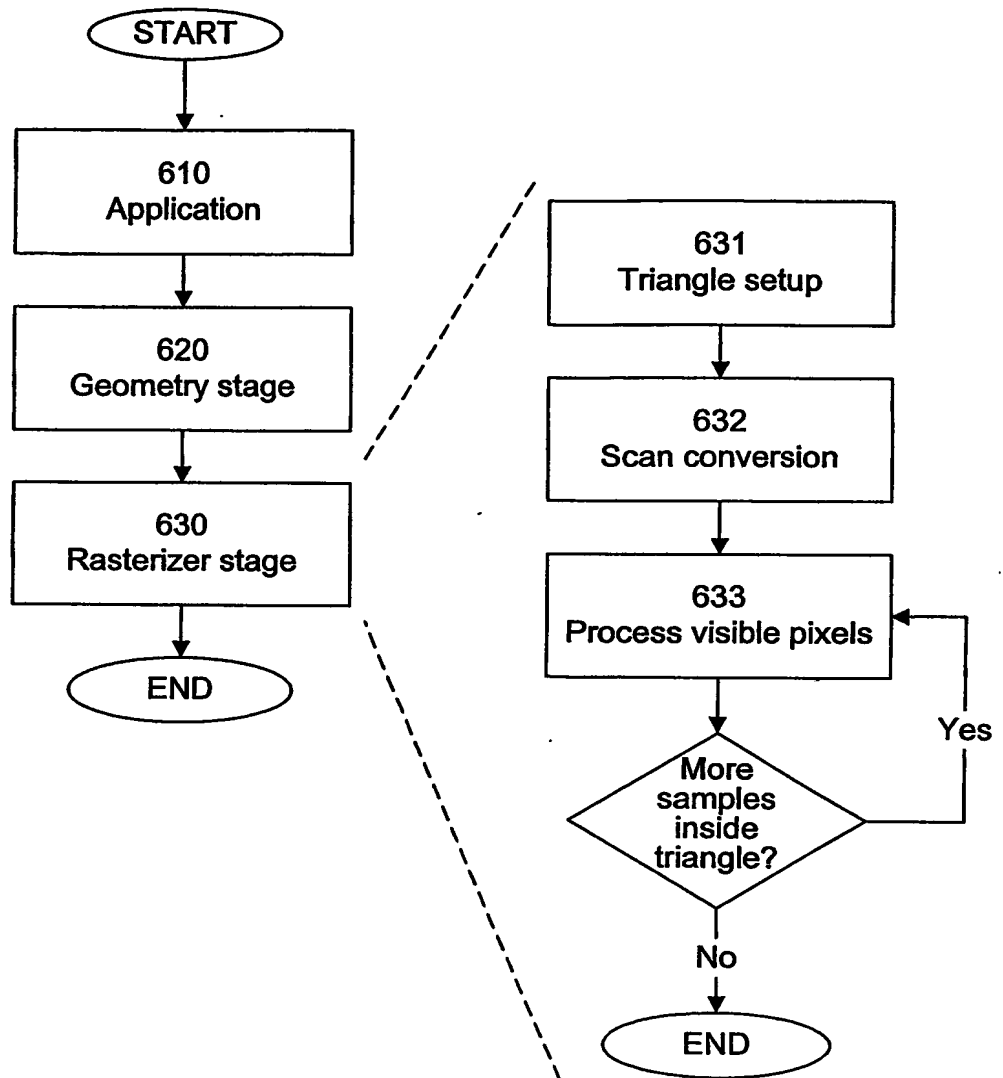
Fig 2b

3/7

*Fig 3*

4/7

*Fig 4*

*Fig 5a**Fig 5b*

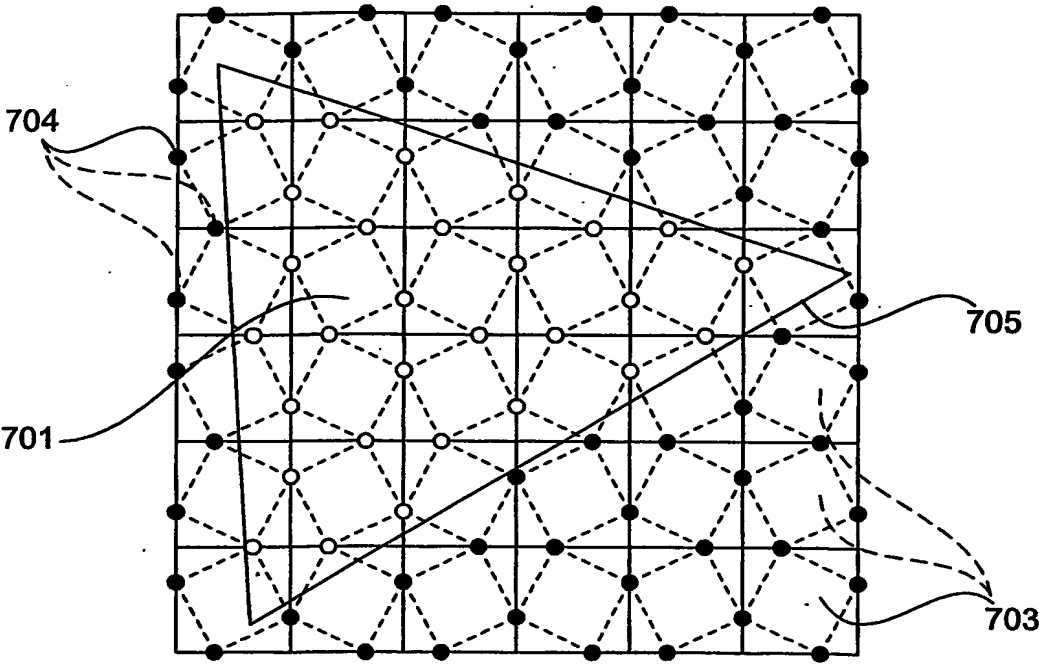


Fig 6a

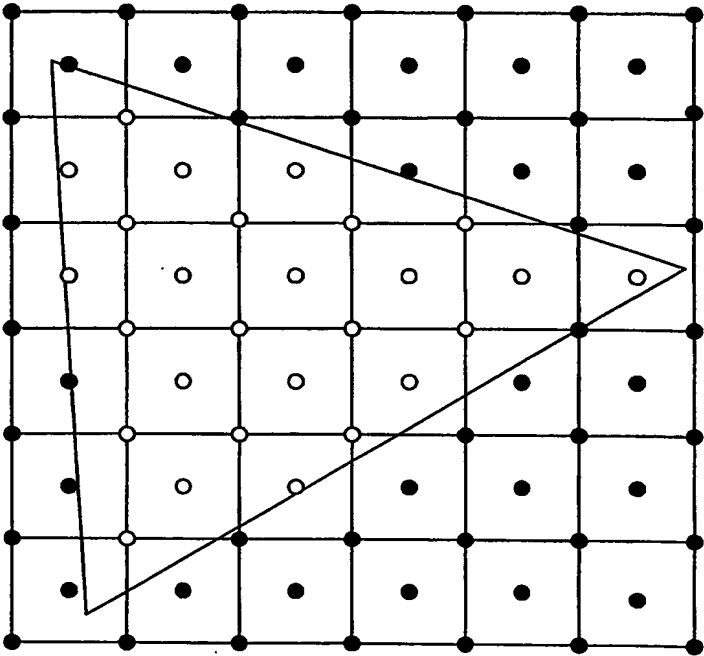


Fig 6b

7/7



Fig 7a



Fig 7b

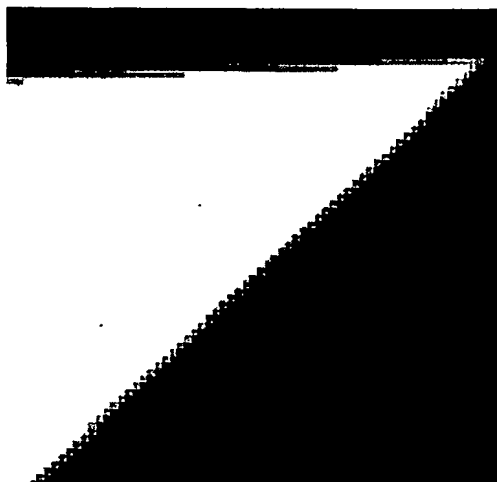


Fig 7c